University of Houston

COSC 6377
Computer Networking

Dr. Omprakash Gnawali

Homework 3

Ashutosh Kumar
1854393

1 Introduction

In this assignment, we are required to study the delay function of an HTTP server. This server delays the `POST` request with a variable `truncid`, which is a number. My `truncid`, as per my UH ID, is 93.

I will be using my `truncid` as well as some other values to try to figure out how the server is delaying the requests.

2 Some Details about the Program

I wrote a program in the Python language. This program sends a `POST` request to the provided IP address (35.224.82.70) at the appropriate port number (5555).

Depending on the endpoint (`/delayme` or `/donotdelayme`), the server either sends a delayed response or sends a response as fast as possible. To study the role of `truncid`, I had the program take 100 samples of latencies with both endpoints. With the data collected, some graphs are generated, which will be used for the analysis.

The Python code is available below.

3 Analysis

Looking at the graphs 1, we can see that there is definitely something that the server is doing with the `truncid`.

For a given `truncid`, we get delayed latencies that are within some range. And this range is different for different values of `truncid`. For instance, for a `truncid` of 93, roughly, the range for the delayed latency is between 140 and 190, while for 92, it is only between 140 and 150. For 91, the range is much larger, being between 150 and 300, roughly. It is possible that this change in the range is simply anecdotal.

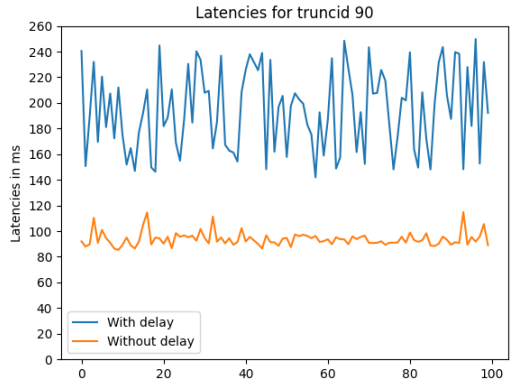
As we move from `truncid` value of 92 to 94, we see that the range increases. Similarly, as we move from 90 to 91, the range increases and then decreases at 92.

It looks as though this range increases until it reaches some threshold and then it resets and starts increasing again. If this hypothesis is true, then `truncid` does not seem to serve any purpose.

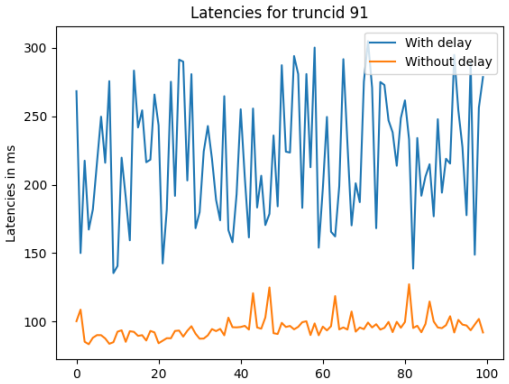
Besides from the above "pattern", I did not find any relation between `truncid` and the delayed latency. The delayed latencies do follow *some* consistency for the same `truncid` when run multiple times.

4 Conclusion

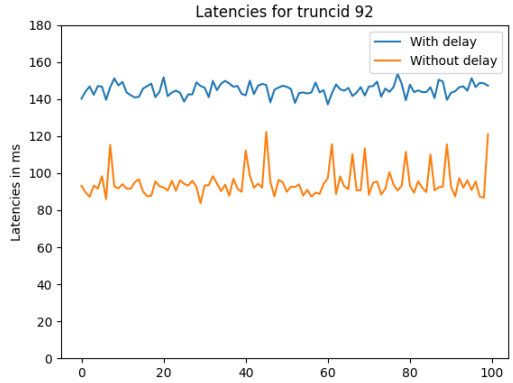
There does not seem to be any pattern between the values of `truncid` and the delayed latency. There, however, is some consistency in the delayed latency. The server may just be following adding some constant latency depending on the value of the `truncid`.



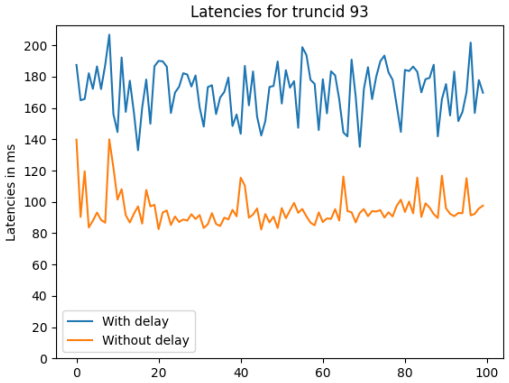
(a) Latency graph for `truncid` 90



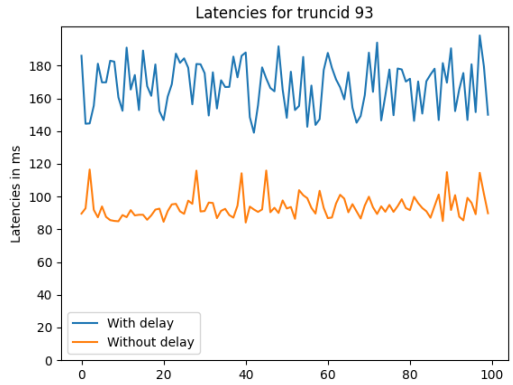
(b) Latency graph for `truncid` 91



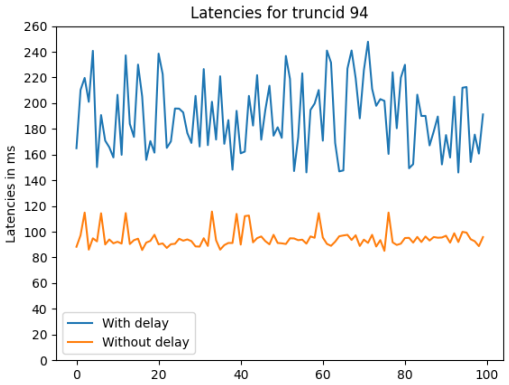
(c) Latency graph for `truncid` 92



(d) Latency graph for `truncid` 93



(e) Latency graph for `truncid` 93



(f) Another latency graph for `truncid` 94

Figure 1: Latencies for a variety of `truncid` values

5 Python Code

```python

import statistics
import time
import requests
import matplotlib.pyplot as plt


def monitor(truncid):
    payload = {"truncid": truncid}

    latencies_with_delay = []
    latencies_without_delay = []

    for i in range(100):
        print(f"{i}->")

        # without delay
        start_time = time.time()
        response = requests.post("http://35.224.82.70:5555/donotdelayme", data=payload)
        end_time = time.time()
        latency_1 = (end_time - start_time) * 1000

        print(f"Response for donotdelayme: {response.json()}")
        print(f"Latency without delay = {latency_1}ms")

        time.sleep(1)

        # with delay
        start_time = time.time()
        response = requests.post("http://35.224.82.70:5555/delayme", data=payload)
        end_time = time.time()
        latency_2 = (end_time - start_time) * 1000

        print(f"Response for delayme: {response.json()}")
        print(f"Latency with delay = {latency_2}ms")
        latencies_without_delay.append(latency_1)
        latencies_with_delay.append(latency_2)

        print("----------------------------")
        time.sleep(1)

    print(f"Latencies without delay: {latencies_without_delay}")
    print(f"Latencies with delay: {latencies_with_delay}")
    print()
    print(f"Avg latency without delay: {statistics.mean(latencies_without_delay)}")
    print(f"Avg latency with delay: {statistics.mean(latencies_with_delay)}")

    plt.figure(truncid)
    plt.plot(latencies_with_delay, label="With delay")
    plt.plot(latencies_without_delay, label="Without delay")
```

```
51
52      plt.legend()
53      plt.title(f"Latencies for truncid {truncid}")
54      plt.ylabel("Latencies in ms")
55      plt.savefig(f"Graphs/{truncid}_latencies.png")
56
57
58  if __name__ == "__main__":
59      monitor(93)
```